IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellants:	Murali Bashyam, Norman W. Finn, Abhijit Patra		
Assignee:	Cisco Technology, Inc.		
Title:	TCP Proxy Connection Management In A Gigabit Environment		
Application No.:	10/051,634	Filing Date:	January 18, 2002
Examiner:	Ramsey Refai	Group Art Unit:	3627
Docket No.:	CIS0139US	Confirmation No.:	8726

Austin, Texas
July 9, 2007

Mail Stop: Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF UNDER 37 CFR § 41.37

Dear Sir:

This brief is timely submitted in support of the Notice of Appeal and Pre-Appeal Brief Request for Review regarding the final rejection of claims 1-124. The appellants note that the Notice of Appeal and Pre-Appeal Brief Request for Review was filed April 23, 2007 (having a two-month extendable period for filing an Appeal Brief), and the subsequent Notice of Panel Decision from Pre-Appeal Brief Review dated June 7, 2007, allows a one-month extendable time period for filing this Appeal Brief, with that time period ending on July 9, 2007 (July 7, 2007 having been a Saturday).

Please charge deposit account No. 502306 for the fee of $500.00 associated with this appeal brief. Please charge this deposit account for any additional sums which may be required to be paid as part of this appeal.

I. REAL PARTY IN INTEREST

The real party in interest on this appeal is Cisco Technology, Inc.

II. RELATED APPEALS AND INTERFERENCES

There are no appeals or interferences related to this application.

III. STATUS OF CLAIMS

Claims 1-124 are pending in the application.

Claims 1-124 stand rejected.

Appellant appeals the rejection of claims 1-124.

IV. STATUS OF AMENDMENTS

No amendments were filed subsequent to the final rejection of November 14, 2006.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 sets forth a method of managing network communication. The method involves terminating a first transmission control protocol ("TCP") connection (e.g., connection 115 of Fig. 1) at a first network element (e.g., TCP Proxy 120 of Fig. 1), wherein said first TCP connection is between said first network element and a second network element (e.g., Client 110 of Fig. 1), and said first TCP connection is intended to be terminated at a third network element (e.g., Server 130 of Fig. 1). See e.g., Specification, p. 5, lines 25-28. The method then involves initiating a second TCP connection (e.g., connection 125) between said first network element and a third network element and establishing communications between said second and said third network elements via said first network element. See e.g., Specification, p. 5, lines 28-30. The method determines a need for data transfer between said second and said third network elements by monitoring an amount of space available in at least one of a plurality of data buffers (e.g., one of transmit buffers 222(1)-222(n) of Fig. 2B). The method then transfers said data between said second and said third network elements. See e.g., Specification, p. 9, lines 18-30, and p. 11, lines 13-33.

Independent claim 32 sets forth a network device (e.g., a TCP Proxy 120 of Fig. 1 and/or TCP Proxy 220 of Figs. 2A and 2B). The network device is configured to terminate a first transmission control protocol ("TCP") connection (e.g., connection 115 of Fig. 1) at a first network element (e.g., a TCP Proxy 120 of Fig. 1 and/or TCP Proxy 220 of Figs. 2A and 2B), wherein said first TCP connection is between said first network element and a second network element (e.g., Client 110 of Fig. 1), and said first TCP connection is intended to be terminated at a third network element (e.g., Server 130 of Fig. 1). See e.g., Specification, p. 5, lines 25-28. The network device is also configured to initiate a second TCP connection (e.g., connection 125 of Fig. 1) between said first network element and a third network element and to establish communications between said second and said third network elements via said first network element. See e.g., Specification, p. 5, lines 28-30. The network device is configured to determine a need for data transfer between said second and said third network elements by monitoring an amount of space available in at least one of a plurality of data buffers (e.g., one of transmit buffers 222(1)-222(n) of Fig. 2B) and to also transfer said data between said

second and said third network elements. See e.g., Specification, p. 9, lines 18-30, and p. 11, lines 13-33.

Independent claim 63 sets forth a network device that includes means (e.g., TCP Proxy 120 of Fig. 1 and/or TCP Proxy 220 of Figs. 2A and 2B, as well as processor 223 and/or network interface 224 of Fig. 2B) for terminating a first transmission control protocol ("TCP") connection (e.g., connection 115 of Fig. 1) at a first network element (e.g., TCP Proxy 120 of Fig. 1 and/or TCP Proxy 220 of Figs. 2A and 2B), wherein said first TCP connection is between said first network element and a second network element (e.g., client 110 of Fig. 1), and said first TCP connection is intended to be terminated at a third network element (e.g., Server 130 of Fig. 1). See e.g., Specification, p. 5, lines 25-28. The network device also includes means (e.g., TCP Proxy 120 of Fig. 1 and/or TCP Proxy 220 of Figs. 2A and 2B, as well as processor 223 and/or network interface 224 of Fig. 2B) for initiating a second TCP connection (e.g., connection 125 of Fig. 1) between said first network element and a third network element. See e.g., Specification, p. 5, lines 28-30. The network device also includes means (e.g., transmit buffers 222(1)-222(n), receive buffers 221(1)-221(n), processor 223, and/or network interface 224 of Fig. 2B) for establishing communications between said second and said third network elements via said first network element. The network device includes means (e.g., TCP Proxy 220, processor 223 and/or network interface 224 of Fig. 2B) for determining need for data transfer between said second and said third network elements by monitoring an amount of space available in at least one of a plurality of data buffers, as well as means (e.g., transmit buffers 222(1)-222(n), receive buffers 221(1)-221(n), processor 223, and/or network interface 224 of Fig. 2B) for transferring said data between said second and said third network elements. See e.g., Specification, p. 9, lines 18-30, and p. 11, lines 13-33.

Independent claim 94 sets forth a computer program product for managing network communication, encoded in computer readable media, said program product comprising a set of instructions executable on a computer system, said set of instructions configured to terminate a first transmission control protocol ("TCP") connection (e.g., connection 115 of Fig. 1) at a first network element (e.g., TCP Proxy 120 of Fig. 1 and/or TCP Proxy 220 of Figs. 2A and 2B), wherein said first TCP connection is between said first network element and a second network element (e.g., Client 110 of Fig. 1), and said

first TCP connection is intended to be terminated at a third network element (e.g., Server 130 of Fig. 1). See e.g., Specification, p. 5, lines 25-28. The instructions are configured to initiate a second TCP connection (e.g., connection 125 of Fig. 1) between said first network element and a third network element, as well as to establish communications between said second and said third network elements via said first network element. See e.g., Specification, p. 5, lines 28-30. The instructions are configured to determine a need for data transfer between said second and said third network elements by monitoring an amount of space available in at least one of a plurality of data buffers (e.g., one of transmit buffers 222(1)-222(n) of Fig. 2B) and to transfer said data between said second and said third network elements. See e.g., Specification, p. 9, lines 18-30, and p. 11, lines 13-33.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

I. Whether claims 1-10, 15-41, 46-72, 77-103, and 108-124 are patentable under 35 U.S.C. §103(a) over Cohen et al, U.S. Patent No. 6,389,462 (hereinafter referred to as "Cohen") in view of Smith et al, U.S. Patent No. 6,308,634 (hereinafter referred to as "Smith").

II. Whether claims 11, 13, 42, 44, 73, 75, 104, and 106 are patentable under 35 U.S.C. §103(a) over Cohen in view of Smith and further in view of Riddle, U.S. Patent No. 5,920,732 (hereinafter referred to as "Riddle").

III. Whether claims 12, 14, 43, 45, 74, 76, 105, and 107 are patentable under 35 U.S.C. §103(a) over Cohen in view of Smith and further in view of Radko, U.S. Patent No. 5,687,392 (hereinafter referred to as "Radko").

VII. ARGUMENT

I. Rejection under 35 U.S.C. §103(a) as being unpatentable over Cohen in view of Smith.

Claim 1 recites:

A method of managing network communication comprising:

terminating a first transmission control protocol ("TCP") connection at a first network element, wherein said first TCP connection is between said first network element and a second network element, and said first TCP connection is intended to be terminated at a third network element;

initiating a second TCP connection between said first network element and a third network element;

establishing communications between said second and said third network elements via said first network element;

determining need for data transfer between said second and said third network elements by monitoring an amount of space available in at least one of a plurality of data buffers; and

transferring said data between said second and said third network elements (emphasis added).

The Examiner relies upon Cohen to teach the operations of "terminating," "initiating," "establishing," and "transferring." Final Office Action (mailed November 14, 2006), pages 2-3. In the rejection, the Examiner equates Cohen's proxy with the first network element, Cohen's client with the second network element, and Cohen's origin server with the third network element.

The Examiner notes that Cohen fails to determine the need for data transfer between the second and third network elements in the manner recited in claim 1. In particular, Cohen fails to determine the need for data transfer "by monitoring an amount of space available in at least one of a plurality of data buffers." The Examiner relies upon col. 13, lines 29-57 of Smith to teach this feature of the claim.

The cited portion of Smith recites:

> FIG. 16 is a flow chart summarizing a method 1600 for writing data to an allocated input or output buffer. Method 1600 will be described with reference to writing data to an allocated input buffer, but is equally well suited to writing server data to an output buffer. In a first step 1602, a client process (e.g., client process 204(1)) uses input buffer identifier 1204(1) to retrieve the buffer status information (the start address 1304 and the length of valid data 1306) for the allocated buffer 1212. Then, in a second step 1604, client process 204(1) transfers a first block of the available client data into the allocated buffer 1212. Client process 204(1) calculates the storage address for the block of data by adding the length of valid data 1306 (data written value) to the start address 1302 of the buffer. Then, in a third step 1606, client process 204(1) updates the buffer status information by incrementing the length of valid data 1306 (data written value) by the size of the data block written to the allocated buffer 1212. Next, in a fourth step 1608, client process 204(1) determines whether the transferred block of data included an end-of-data indicator, and if so then method 1600 ends.
>
> If, in fourth step 1608, client process 204(1) determines that the transferred data block did not include an end-of-file indicator, then in a fifth step 1610 client process 204(1) determines whether the allocated buffer is full by comparing the updated length of valid data 1306 to the known size of buffer 1212. If the data buffer 1212 is not full, then method 1600 returns to second step 1604 to transfer the next block of data. Smith, col. 13, lines 29-57.

The above-quoted section of Smith teaches how data can be written into a buffer. The first steps, which write a block of data to the buffer, are described as being performed unconditionally. Then, two determinations are made: the client process determines whether the transferred data block includes an end-of-data / end-of-file indicator at 1608, and the client process determines whether the buffer is full at 1610.

The only determination in the cited portion of Smith that can arguably be said to determine any <u>need</u> to transfer data is the determination that is made based upon the presence or lack of an end-of-data indication in the just-written data block. In contrast, the determination based upon the fullness of the buffer is made to determine whether data <u>can</u> be transferred into the buffer. The fullness of the buffer is in no way used to determine the need to transfer data; instead, this factor only controls whether data can be transferred into the buffer. This is highlighted by the fact that the buffer fullness determination (1610) is only made if no end-of-data indicator is found (1608). *See* Smith, Fig. 16 (showing that the buffer fullness determination is only made in situations in

which a need to transfer additional data has already been identified based upon the lack of the end-of-data indicator). Thus, for at least this reason, the cited portion of Smith clearly fails to teach or suggest "determining need for data transfer between said second and said third network elements by monitoring an amount of space available in at least one of a plurality of data buffers."

In response to the above arguments, the Examiner states: "Smith teaches that the process determines that the transferred data block did not include an end-of file indicator prior to determining if the buffer is full. Meaning, the process recognizes that more data is still to come (need to transfer). The process checks the status of the buffer and transfers data if the buffer is not full." Advisory Action mailed February 21, 2007, p. 2. Appellants again note that this simply summarizes the cited portions of Smith, which teach checking the status of the buffer only after it has already been determined, based on the lack of an end-of-file indicator, that there is a need for more data transfer. Checking the status of the buffer allows the process to determine whether more data can be transferred to satisfy the already-identified need for data transfer. Determining whether data can be transferred is clearly not the same as determining whether there is a need to transfer data. Thus, the cited portions of Smith still clearly fail to teach or suggest the features of claim 1.

Furthermore, Appellants note that the cited portions of Smith simply teach writing data, which appears to already be available locally, to a buffer. Detecting the presence or absence of the end-of-data indication is only used to determine whether there is more data to write to the buffer, not to determine "need for data transfer between said second and said third network elements," as recited in claim 1. Accordingly, the cited portion of Smith also clearly fails to teach or suggest this feature of the claim.

Even if Smith and Cohen are combined in the manner suggested by the Examiner, the resulting combination still fails to teach or suggest the features of claim 1. In particular, the combination of Smith and Cohen would result in a system in which Cohen's proxy or origin server implements the buffer-writing technique described in the cited portion of Smith. Accordingly, such a system would determine the need to write additional server data, which is already available at the origin server or proxy, into a

buffer based upon whether a previously-written unit of data contained an end-of-data indicator. This system would clearly not determine the need to transfer data between the origin server and a client, nor would this system make such a determination based upon an amount of space available in a buffer.

For at least the foregoing reasons, the cited art fails to teach or suggest "determining need for data transfer between said second and said third network elements by monitoring an amount of space available in at least one of a plurality of data buffers." Claim 1 and dependent claims 2-10 and 15-31 are patentable over the cited art for at least the foregoing reasons. Claims 32-41, 46-72, 77-103, and 108-124 are patentable over the cited art for similar reasons and are thus grouped with claim 1 for purposes of this appeal.

<u>II. Rejection under 35 U.S.C. §103(a) as being unpatentable over Cohen in view of Smith and in further view of Riddle.</u>

Claims 11, 13, 42, 44, 73, 75, 104, and 106 stand rejected under 35 U.S.C. §103(a), as being unpatentable over Cohen in view of Smith and in further view of Riddle. These claims are patentable over the cited art for reasons similar to those presented above with respect to claim 1 and are thus grouped with claim 1 for purposes of this appeal.

<u>III. Rejection under 35 U.S.C. §103(a) as being unpatentable over Cohen in view of Smith and further in view of Radko.</u>

Claims 12, 14, 43, 45, 74, 76, 105, and 107 stand rejected under 35 U.S.C. §103(a), as being unpatentable over Cohen, in view of Smith and in further view of Radko. These claims are patentable over the cited art for reasons similar to those presented above with respect to claim 1 and are thus grouped with claim 1 for purposes of this appeal.

CONCLUSION

The appellants respectfully submit that claims 1-124 are allowable over the cited references for at least the above-stated reasons. The appellants respectfully request that the Board reverse the rejections of these claims.

Respectfully submitted,

Brenna A Brock

Brenna A. Brock
Attorney for Appellant
Reg. No. 48,509
512-439-5087
512-439-5099 (fax)

VIII. CLAIMS APPENDIX

1. (Previously Presented) A method of managing network communication comprising:

terminating a first transmission control protocol ("TCP") connection at a first network element, wherein said first TCP connection is between said first network element and a second network element, and said first TCP connection is intended to be terminated at a third network element;

initiating a second TCP connection between said first network element and a third network element;

establishing communications between said second and said third network elements via said first network element;

determining need for data transfer between said second and said third network elements by monitoring an amount of space available in at least one of a plurality of data buffers; and

transferring said data between said second and said third network elements.

2. (Original) The method of claim 1, wherein said second network element initiates said first TCP connection for said third network element.

3. (Original) The method of claim 1, wherein said communications between said second and said third network elements are established using said first and said second TCP connections.

4. (Original) The method of claim 1, wherein said communications between said second and said third network elements forms an end-to-end TCP connection.

5. (Original) The method of claim 1, wherein said first network element is a proxy server.

6. (Original) The method of claim 1, wherein a control unit of said proxy server monitors said plurality of buffers.

7. (Original) The method of claim 1, wherein said control unit transfers said data between said second and said third network elements.

8. (Original) The method of claim 1, wherein said proxy server supports transparent communications between said second and said third network elements.

9. (Previously Presented) The method of claim 1, wherein at least one of said plurality of data buffers is a receive buffer.

10. (Previously Presented) The method of claim 1, wherein at least one of said plurality of data buffers is a transmit buffer.

11. (Previously Presented) The method of claim 9, wherein said receive buffer is pre-allocated.

12. (Previously Presented) The method of claim 9, wherein said receive buffer is dynamically allocated.

13. (Original) The method of claim 10, wherein said transmit buffer is pre-allocated.

14. (Original) The method of claim 10, wherein said transmit buffer is dynamically allocated.

15. (Original) The method of claim 1, wherein said second network element is one of a plurality of clients.

16. (Original) The method of claim 1, wherein one of a plurality of applications on said client initiates said first TCP connection for said client.

17. (Original) The method of claim 1, wherein said third network element is one of a plurality of servers.

18. (Original) The method of claim 1, wherein a data switching unit of said proxy server determines which one of said plurality of servers to use for said second TCP connection.

19. (Original) The method of claim 1, further comprising:
monitoring said first TCP connection.

20. (Original) The method of claim 19, further comprising:
receiving a request for data from said application; and
determining whether said request requires said second TCP connection with one of said plurality of servers.

21. (Original) The method of claim 20, wherein data switching unit receives said request for data via said control unit.

22. (Original) The method of claim 20, wherein said determining of said second TCP connection is done by said data switching unit.

23. (Original) The method of claim 20, further comprising:
if said request does not require said second TCP connection with one of said plurality of servers,
servicing said request for data, and
closing said connection with said client.

24. (Original) The method of claim 23, wherein said request for data is served by passing data from said data switching unit to said control unit for transmission to said application on said client.

25. (Original) The method of claim 23, further comprising:
if said request requires said second TCP connection with one of said plurality of servers,
selecting a first server from said plurality of servers, and
initiating said second TCP connection with said first server.

26. (Original) The method of claim 25, wherein said application requests said end-to-end TCP connection with said first server.

27. (Previously Presented) The method of claim 25, further comprising:
receiving said data on said second TCP connection from said first server;
storing said data in said receive buffer of said second TCP connection;
transferring said data from said receive buffer to said transmit buffer of said first TCP connection;
monitoring space in said transmit buffer; and
if said transmit buffer has space,
determining whether said first TCP connection needs additional data.

28. (Previously Presented) The method of claim 27, further comprising:
if said first TCP connection needs said additional data,
requesting said additional data from said first server; and
repeating said steps of receiving, storing, transferring, monitoring and determining until said request for data from said application is served.

29. (Original) The method of claim 28, wherein said additional data is transferred into said transmit buffer without a request for said additional data.

30. (Original) The method of claim 28, further comprising:
if said request for data from said application is served,
closing said first TCP connection with said client.

31. (Original) The method of claim 30, wherein said closing of said connection is done by said control unit upon a receiving a request for closing said connection from said data switching unit.

32. (Previously Presented) A network device comprising:

terminate a first transmission control protocol ("TCP") connection at a first network element, wherein said first TCP connection is between said first network element and a second network element, and said first TCP connection is intended to be terminated at a third network element;

initiate a second TCP connection between said first network element and a third network element;

establish communications between said second and said third network elements via said first network element;

determine need for data transfer between said second and said third network elements by monitoring an amount of space available in at least one of a plurality of data buffers; and

transfer said data between said second and said third network elements.

33. (Original) The network device of claim 32, wherein said second network element initiates said first TCP connection for said third network element.

34. (Original) The network device of claim 32, wherein said communications between said second and said third network elements are established using said first and said second TCP connections.

35. (Original) The network device of claim 32, wherein said communications between said second and said third network elements forms an end-to-end TCP connection.

36. (Original) The network device of claim 32, wherein said first network element is a proxy server.

37. (Original) The network device of claim 32, wherein a control unit of said proxy server monitors said plurality of buffers.

38. (Original) The network device of claim 32, wherein said control unit transfers said data between said second and said third network elements.

39. (Original) The network device of claim 32, wherein said proxy server supports transparent communications between said second and said third network elements.

40. (Original) The network device of claim 32, wherein at least one of said plurality of buffers is a receive buffer.

41. (Original) The network device of claim 32, wherein at least one of said plurality of buffers is a transmit buffer.

42. (Previously Presented) The network device of claim 40, wherein said receive buffer is pre-allocated.

43. (Previously Presented) The network device of claim 40, wherein said receive buffer is dynamically allocated.

44. (Original) The network device of claim 41, wherein said transmit buffer is pre-allocated.

45. (Original) The network device of claim 41, wherein said transmit buffer is dynamically allocated.

46. (Original) The network device of claim 32, wherein said second network element is one of a plurality of clients.

47. (Original) The network device of claim 32, wherein one of a plurality of applications on said client initiates said first TCP connection for said client.

48. (Original) The network device of claim 32, wherein said third network element is one of a plurality of servers.

49. (Original) The network device of claim 32, wherein a data switching unit of said proxy server determines which one of said plurality of servers to use for said second TCP connection.

50. (Original) The network device of claim 32, wherein said processor is further configured to monitor said first TCP connection.

51. (Original) The network device of claim 50, wherein said processor is further configured to receive a request for data from said application; and determine whether said request requires said second TCP connection with one of

said plurality of servers.

52. (Original) The network device of claim 51, wherein data switching unit receives said request for data via said control unit.

53. (Original) The network device of claim 51, wherein said determining of said second TCP connection is done by said data switching unit.

54. (Original) The network device of claim 51, wherein said processor is further configured to if said request does not require said second TCP connection with one of said plurality of servers,

service said request for data, and

close said connection with said client.

55. (Original) The network device of claim 54, wherein said request for data is served by passing data from said data switching unit to said control unit for transmission to said application on said client.

56. (Original) The network device of claim 54, wherein said processor is further configured to if said request requires said second TCP connection with one of said plurality of servers,

select a first server from said plurality of servers, and

initiate said second TCP connection with said first server.

57. (Original) The network device of claim 56, wherein said application requests said end-to-end TCP connection with said first server.

58. (Previously Presented) The network device of claim 56, wherein said processor is further configured to receive said data on said second TCP connection from said first server;

store said data in said receive buffer of said second TCP connection;

transfer said data from said receive buffer to said transmit buffer of said first TCP connection;

monitor space in said transmit buffer; and

if said transmit buffer has space,

determine whether said first TCP connection needs additional data.

59. (Previously Presented) The network device of claim 58, wherein said processor is further configured to if said first TCP connection needs said additional data,

request said additional data from said first server; and

repeat said steps of receiving, storing, transferring, monitoring and determining until said request for data from said application is served.

60. (Original) The network device of claim 59, wherein said additional data is transferred into said transmit buffer without a request for said additional data.

61. (Original) The network device of claim 59, wherein said processor is further configured to if said request for data from said application is served,
close said first TCP connection with said client.

62. (Original) The network device of claim 61, wherein said closing of said connection is done by said control unit upon a receiving a request for closing said connection from said data switching unit.

63. (Previously Presented) A network device comprising:
means for terminating a first transmission control protocol ("TCP") connection at a first network element, wherein said first TCP connection is between said first network element and a second network element, and said first TCP connection is intended to be terminated at a third network element;
means for initiating a second TCP connection between said first network element and a third network element;
means for establishing communications between said second and said third network elements via said first network element;
means for determining need for data transfer between said second and said third network elements by monitoring an amount of space available in at least one of a plurality of data buffers; and
means for transferring said data between said second and said third network elements.

64. (Original) The network device of claim 63, wherein said second network element initiates said first TCP connection for said third network element.

65. (Original) The network device of claim 63, wherein said communications between said second and said third network elements are established using said first and said second TCP connections.

66. (Original) The network device of claim 63, wherein said communications between said second and said third network elements forms an end-to-end TCP connection.

67. (Original) The network device of claim 63, wherein said first network element is a proxy server.

68. (Original) The network device of claim 63, wherein a control unit of said proxy server monitors said plurality of buffers.

69. (Original) The network device of claim 63, wherein said control unit transfers said data between said second and said third network elements.

70. (Original) The network device of claim 63, wherein said proxy server supports transparent communications between said second and said third network elements.

71. (Original) The network device of claim 63, wherein at least one of said plurality of buffers is a receive buffer.

72. (Previously Presented) The network device of claim 63, wherein at least one of said plurality of buffers is a transmit buffer.

73. (Previously Presented) The network device of claim 71, wherein said receive buffer is pre-allocated.

74. (Previously Presented) The network device of claim 71, wherein said receive buffer is dynamically allocated.

75. (Original) The network device of claim 72, wherein said transmit buffer is pre-allocated.

76. (Original) The network device of claim 72, wherein said transmit buffer is dynamically allocated.

77. (Previously Presented) The network device of claim 63, wherein said second network element is one of a plurality of clients.

78. (Previously Presented) The network device of claim 63, wherein one of a plurality of applications on said client initiates said first TCP connection for said client.

79. (Previously Presented) The network device of claim 63, wherein said third network element is one of a plurality of servers.

80. (Previously Presented) The network device of claim 63, wherein a data switching unit of said proxy server determines which one of said plurality of servers to use for said second TCP connection.

81. (Previously Presented) The network device of claim 63, further comprising:

means for monitoring said first TCP connection.

82. (Original) The network device of claim 81, further comprising:

means for receiving a request for data from said application; and

means for determining whether said request requires said second TCP connection with one of said plurality of servers.

83. (Original) The network device of claim 82, wherein data switching unit receives said request for data via said control unit.

84. (Original) The network device of claim 82, wherein said determining of said second TCP connection is done by said data switching unit.

85. (Original) The network device of claim 82, further comprising:
means for servicing said request for data if said request does not require said second TCP connection with one of said plurality of servers; and
means for closing said connection with said client if said request does not require said second TCP connection with one of said plurality of servers.

86. (Original) The network device of claim 85, wherein said request for data is served by passing data from said data switching unit to said control unit for transmission to said application on said client.

87. (Original) The network device of claim 85, further comprising:
means for selecting a first server from said plurality of servers if said request requires said second TCP connection with one of said plurality of servers; and
means for initiating said second TCP connection with said first server if said request requires said second TCP connection with one of said plurality of servers.

88. (Original) The network device of claim 87, wherein said application requests said end-to-end TCP connection with said first server.

89. (Previously Presented) The network device of claim 87, further comprising:
means for receiving said data on said second TCP connection from said first server;
means for storing said data in said receive buffer of said second TCP connection;
means for transferring said data from said receive buffer to said transmit buffer of said first TCP connection;
means for monitoring space in said transmit buffer; and
means for determining whether said first TCP connection needs additional data if said transmit buffer has space.

90. (Previously Presented) The network device of claim 89, further comprising:

means for requesting said additional data from said first server if said first TCP connection needs said additional data;

means for repeating said steps of receiving, storing, transferring, monitoring if said first TCP connection needs said additional data; and

means for determining until said request for data from said application is served.

91. (Original) The network device of claim 90, wherein said additional data is transferred into said transmit buffer without a request for said additional data.

92. (Original) The network device of claim 90, further comprising:

means for closing said first TCP connection with said client if said request for data from said application is served.

93. (Original) The network device of claim 92, wherein said closing of said connection is done by said control unit upon a receiving a request for closing said connection from said data switching unit.

94. (Previously Presented) A computer program product for managing network communication, encoded in computer readable media, said program product comprising a set of instructions executable on a computer system, said set of instructions configured to

terminate a first transmission control protocol ("TCP") connection at a first network element, wherein said first TCP connection is between said first network element and a second network element, and said first TCP connection is intended to be terminated at a third network element;

initiate a second TCP connection between said first network element and a third network element;

establish communications between said second and said third network elements via said first network element;

determine need for data transfer between said second and said third network

elements by monitoring an amount of space available in at least one of a plurality of data buffers; and

transfer said data between said second and said third network elements.

95. (Original) The computer program product of claim 94, wherein said second network element initiates said first TCP connection for said third network element.

96. (Original) The computer program product of claim 94, wherein said communications between said second and said third network elements are established using said first and said second TCP connections.

97. (Original) The computer program product of claim 94, wherein said communications between said second and said third network elements forms an end-to-end TCP connection.

98. (Original) The computer program product of claim 94, wherein said first network element is a proxy server.

99. (Original) The computer program product of claim 94, wherein a control unit of said proxy server monitors said plurality of buffers.

100. (Original) The computer program product of claim 94, wherein said control unit transfers said data between said second and said third network elements.

101. (Original) The computer program product of claim 94, wherein said proxy server supports transparent communications between said second and said third network elements.

102. (Original) The computer program product of claim 94, wherein at least one of said plurality of buffers is a receive buffer.

103. (Original) The computer program product of claim 94, wherein at least one of said plurality of buffers is a transmit buffer.

104. (Original) The computer program product of claim 102, wherein said receive buffer is pre-allocated.

105. (Original) The computer program product of claim 102, wherein said receive buffer is dynamically allocated.

106. (Original) The computer program product of claim 103, wherein said transmit buffer is pre-allocated.

107. (Original) The computer program product of claim 103, wherein said transmit buffer is dynamically allocated.

108. (Original) The computer program product of claim 94, wherein said second network element is one of a plurality of clients.

109. (Original) The computer program product of claim 94, wherein one of a plurality of applications on said client initiates said first TCP connection for said client.

110. (Original) The computer program product of claim 94, wherein said third network element is one of a plurality of servers.

111. (Original) The computer program product of claim 94, wherein a data switching unit of said proxy server determines which one of said plurality of servers to use for said second TCP connection.

112. (Original) The computer program product of claim 94, wherein said set of instructions is further configured to:
monitor said first TCP connection.

113. (Original) The computer program product of claim 112, wherein said set of instructions is further configured to:
receive a request for data from said application; and
determine whether said request requires said second TCP connection with one of said plurality of servers.

114. (Original) The computer program product of claim 113, wherein data switching unit receives said request for data via said control unit.

115. (Original) The computer program product of claim 113, wherein said determining of said second TCP connection is done by said data switching unit.

116. (Previously Presented) The computer program product of claim 113, wherein said set of instructions is further configured to:
if said request does not require said second TCP connection with one of said plurality of servers,
service said request for data, and
close said connection with said client.

117. (Original) The computer program product of claim 116, wherein said request for data is served by passing data from said data switching unit to said control unit for transmission to said application on said client.

118. (Original) The computer program product of claim 116, wherein said set of instructions is further configured to :
if said request requires said second TCP connection with one of said plurality of servers,
select a first server from said plurality of servers, and
initiate said second TCP connection with said first server.

119. (Original) The computer program product of claim 118, wherein said application requests said end-to-end TCP connection with said first server.

120. (Previously Presented) The computer program product of claim 118, wherein said set of instructions is further configured to:
receive said data on said second TCP connection from said first server;
store said data in said receive buffer of said second TCP connection;
transfer said data from said receive buffer to said transmit buffer of said first TCP connection;
monitor space in said transmit buffer; and
if said transmit buffer has space,
determine whether said first TCP connection needs additional data.

121. (Previously Presented) The computer program product of claim 120, wherein said set of instructions is further configured to:
if said first TCP connection needs said additional data,
request said additional data from said first server; and
repeat said steps of receiving, storing, transferring, monitoring and determining until said request for data from said application is served.

122. (Original) The computer program product of claim 121, wherein said additional data is transferred into said transmit buffer without a request for said additional data.

123. (Original) The computer program product of claim 121, wherein said set of instructions is further configured to :
if said request for data from said application is served,
close said first TCP connection with said client.

124. (Original) The computer program product of claim 123, wherein said closing of said connection is done by said control unit upon a receiving a request for closing said connection from said data switching unit.

IX. EVIDENCE APPENDIX

None

X. RELATED PROCEEDINGS APPENDIX

None